INRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France

Tél.(1) 39 63 55 11

Rapports de Recherche

N° 549

ON MOTION COORDINATION

Pierre TOURNASSOUD

Juillet 1986

July 1986

On Motion Coordination

Pierre Tournassoud

INRIA

Domaine de Voluceau-Rocquencourt

BP 105, 78153 Le Chesnay FRANCE

Résumé

Pour réaliser une cellule robotique avec plusieurs manipulateurs partageant le même espace de travail, il faut disposer d'un module qui coordonne leurs mouvements et prévient les collisions. Nous nous limitons ici à une coordination locale des déplacements, faisant l'hypothèse que l'environnement n'est pas trop encombré là où les robots interfèrent.

Nous définissons tout d'abord des outils pour analyser les déplacements de plusieurs mobiles. Nous posons ensuite le problème de leur coordination comme un problème local de minimisation, qui se traduit par des contraintes géométriques naturelles sur la tâche: les trajectoires des mobiles sont séparées en les forçant à glisser sur des hyperplans tangents séparateurs.

A l'opposée des méthodes globales dont la complexité exponentielle impose de réduire arbitrairement le nombre de degrés de liberté de la tâche, cette approche permet de mettre à profit chacun d'entre eux. Les priorités peuvent être considérées comme des paramètres variables pour augmenter l'espace des solutions.

L'algorithme proposé est un premier pas vers la réalisation d'un module temps-réel de coordination de deux robots. Des résultats de simulation sont présentés pour des manipulateurs de description générale.

Abstract

A necessary step in the realization of a robotic cell with several robots sharing common workspace is the design of an anti-collision module. Instead of addressing the problem of global planning of multiple arms, we limit ourselves to the problem of local coordination of movements, assuming the environment is not too constrained at places where the arms will cross.

We begin by giving some tools for the analysis of the movements of several objects. We then express motion coordination as a local optimization problem, which naturally translates into geometric constraints on the movements. The idea behind separation of trajectories we perform is that of forcing mobile objects to slide on tangent separating hyperplanes.

As opposed to global methods whose inherent exponential complexity impose to arbitrarily reduce the number of degrees of freedom of the task, our approach permits to effectively take advantage of them all. Variable priorities can be used to augment the research space.

The proposed algorithm is a first step towards the realization of a module for on-line coordination of two arms. Results of a simulation are presented for manipulators of general description.

1. Introduction

The problem of coordinating the motions of several mobile objects has been tackled from two opposite points of view. The first one is that of computing on-line modifications of trajectories preventing collision. The other one is that of off-line global planning. It is known as the *Findpath Problem.*

1.1. On Global Planning

1.1.1. The Case of a Single Moving Object

The Findpath Problem has given rise to a number of practical algorithms in the case of a single moving object (say a mobile robot) or one articulated arm with few degrees of freedom. It can be stated as follows: given an initial and a goal configuration, and a description of obstacles in real space, compute a path between both configurations avoiding obstacles, if one exists. The basic idea behind most of proposed algorithms is to express the constraints generated by obstacles in the *Configuration Space* of the moving object, chosen among the spaces representing its degrees of freedom with fewest parameters [Udupa 1977; Lozano-Pérez 1981, 1983].

Nevertheless, as the complexity of the Findpath Problem is inherently exponential in the number of degrees of freedom, this introduces a practical bound. Indeed, for one moving object with translational and rotational degrees of freedom in the plane, Lozano-Pérez slices the range of rotations and comes back to a two-dimensional problem. Faverjon builds an octree to represent free space for the first three degrees of freedom, while a heuristic approach is used for the hand [Faverjon 1984, 1986].

1.1.2. The Case of Multiple Moving Objects

For more than one moving object, we could think of working in a Configuration Space representing all degrees of freedom involved. Nevertheless the limitations stated above make this impossible but for trivial cases. We know so far of two algorithms for path-planning involving multiple moving objects.

The first one [Kant and Zucker 1986] limits itself to the problem of one mobile robot in a planar environment of stationary obstacles and moving obstacles, this is, objects whose trajectories cannot be modified. It decomposes the problem into two sub-problems: first planning a path avoiding collisions with stationary obstacles, then computing the velocity along this path so as to avoid collisions with moving obstacles.

The second algorithm [Erdmann and Lozano-Pérez 1986] tackles the more complex problem of simultaneous planning of many mobile objects, each one having two degrees of freedom. The complexity is reduced by assigning priorities, motions being planned one object at a time. It relies on the idea of *Configuration Space-Time*, represented by a discrete collection of two-dimensional Configuration Spaces at certain points in time.

The interesting idea behind both works is that it is fruitful to explicitely take *time* into account in the context of multiple moving objects. This leads to *Space-Time Planning* algorithms. To make the problem tractable, both algorithms nevertheless limit themselves to planning motion one object at a time, thus restricting the solution space.

1.1.3. Complexity Results

Using a cylindrical algebraic cell decomposition of free space, Schwartz and Sharir reduce the Findpath Problem to Tarski's algorithm for deciding statements in the quantified elementary theory of real numbers, which gives an algorithm polynomial in the number of algebraic constraints and exponential in the number of degrees of freedom [Schwartz and Sharir 1982, 1983].

The particular case of motion coordination of many discs in the plane has given rise to the idea of *retraction* onto the boundary of free space, leading to $O(n^2)$ and $O(n^3)$ time algorithms respectively for two and three discs, where n is the number of walls of the polygonal obstacles [Yap 1984]. Wether there exists an $O(n^k)$ algorithm for the general case of k discs still remains an open question.

In the case of coordinated motion planning of two planar arms modeled on the Stanford arm, an $O(n^3)$ algorithm has been proposed [Fortune, Wilfong and Yap 1986].

We cite among other interesting results: the problem of coordinated motion of many discs is NP-hard in the number of discs [Spirakis and Yap 1983]; the problem of coordinated motion of many rectangles in a rectangular frame is PSPACE-hard [Hopcroft, Schwartz and Sharir 1984].

In short, the inherent exponential complexity of the Find-Path problem makes the proposed algorithms impractical but for few degrees of freedom. On the other hand, practical algorithms for many objects coordination arbitrarily reduce the degrees of freedom of the problem, thus implying the loss of completeness.

1.2. Local Coordination Versus Global Planning

We propose here a *local* method of computing coordinated motion of many moving objects. We take as input joint variable values and desired joint velocities, and compute modified joint velocities preventing collision. The approach is based on the use of *tangent separating hyperplanes* in real space [Tournassoud 1986].

Known local approaches to this problem (see for example [Freund and Hoyer 1984]) tend to rely on a specific description of manipulators. On the contrary, we give tools of general application. A positive point of our algorithm is that it permits to take advantage of every degree of freedom of the problem. A weak point, as for other local approaches, is the difficulty to prove that we will actually exhibit a path between initial and final configurations if one exists.

2. Multiple Moving Object Coordination

The basic idea of this work is to express movement coordination as a local optimization problem. We first introduce some tools for the position of the problem.

Definition: *Let* **n** *be a unit vector and* **o** *a reference point. For a solid* S, *assumed to be strictly convex,* the first point of S in the direction **n** *is the point* **x(n)** *which minimizes the inner product* **ox.n**.

Let $\mathbf{x}_2(\mathbf{n}_{12})$ be the first point of S_2 in direction $\mathbf{n}_{12}$ and $\mathbf{x}_1(\mathbf{n}_{21})$ the first point of S_1 in opposite direction $\mathbf{n}_{21} = -\mathbf{n}_{12}$. There exists a hyperplane of normal $\mathbf{n}_{12}$ separating solids S_1 and S_2 if and only if $[\mathbf{ox}_2(\mathbf{n}_{12}) - \mathbf{ox}_1(\mathbf{n}_{21})].\mathbf{n}_{12} \geq 0$ (Figure 1). In that case $\mathbf{n}_{12}$ points from the side of S_1 towards the side of S_2. Vector $\mathbf{ox}_2(\mathbf{n}_{12}) - \mathbf{ox}_1(\mathbf{n}_{21})$ will also be written $\mathbf{x}_1\mathbf{x}_2$ for short, but we must remain aware that points $\mathbf{x}_1$ and $\mathbf{x}_2$ depend on the choice of the normal. In the sequel, we will often speak of "the" separating hyperplane of normal $\mathbf{n}_{12}$, though there is in fact a family of such hyperplanes, namely those containing a point $\mathbf{x}$ such that $\mathbf{x}_1\mathbf{x}_2.\mathbf{n}_{12} \geq \mathbf{x}_1\mathbf{x}.\mathbf{n}_{12} \geq 0$.

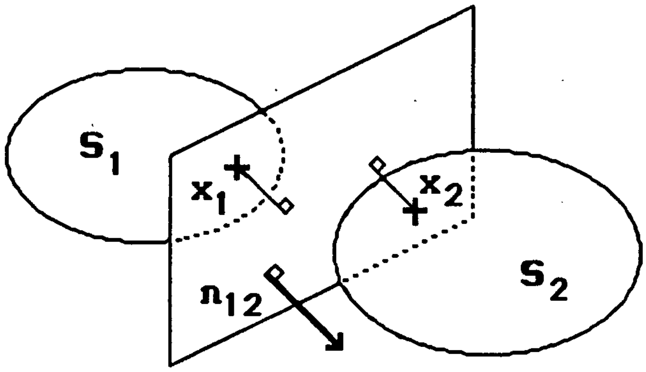


Fig. 1. Two solids and a separating plane

2.1. The Case of Solids in Translation

We first consider the case of two solids translating with respective velocities $\mathbf{v}_1$ and $\mathbf{v}_2$. In a frame attached to S_1 the relative velocity of S_2 is $\mathbf{v}_r=\mathbf{v}_2-\mathbf{v}_1$. We will thus *anticipate* any future collision by forcing: $\mathbf{v}_r.\mathbf{n}_{12} \geq 0$.

We now translate the avoidance problem into the local following optimization problem (P): minimize the perturbation of desired velocities $\underline{\mathbf{v}}_1$ and $\underline{\mathbf{v}}_2$ with the constraints of *separation of solids* (i) and of *separation of trajectories* (ii). Variables of the optimization problem are the unit vector normal to the hyperplane, $\mathbf{n}_{12}$, and the velocities of the solids, $\mathbf{v}_1$ and $\mathbf{v}_2$.

(P) minimize $\|\mathbf{v}_1-\underline{\mathbf{v}}_1\|^2 + \|\mathbf{v}_2-\underline{\mathbf{v}}_2\|^2$

(i) $\mathbf{x}_1\mathbf{x}_2 .\mathbf{n}_{12} \geq 0$

(ii) $(\mathbf{v}_2-\mathbf{v}_1).\mathbf{n}_{12} \geq 0$

The proofs of the two following propositions are given in Appendix I.

Proposition 1: *for constant velocities* $\underline{v}_1$ *and* $\underline{v}_2$, *objects do not collide if and only if there exists a separating hyperplane such that* $(\underline{v}_2-\underline{v}_1).\mathbf{n}_{12} \geq 0$, *else:*

- *the best separating hyperplane is* tangent *to both objects,*

- *the optimal relative velocity lies in the hyperplane.*

Proposition 2: *the global optimization problem* (P) *is equivalent to subproblems* (SP1) *and* (SP2).

(SP1) *for the choice of the optimal normal to the separating hyperplane,* $\mathbf{n}_{12}^*$:

with $\underline{v}_r = \underline{v}_2-\underline{v}_1$,

maximize $\underline{v}_r.\mathbf{n}_{12}$ under the constraint of separation of solids

$\mathbf{x}_1\mathbf{x}_2 .\mathbf{n}_{12} \geq 0$.

(SP2) *for the choice of optimal velocities* $\mathbf{v}_1$ *and* $\mathbf{v}_2$:

minimize $||\mathbf{v}_1-\underline{v}_1||^2 + ||\mathbf{v}_2-\underline{v}_2||^2$ under the constraint of separation of trajectories $(\mathbf{v}_2-\mathbf{v}_1).\mathbf{n}_{12}^* \geq 0$.

Subproblem (SP1) states that the three vectors ($\underline{v}_r$, $\mathbf{x}_1\mathbf{x}_2$, $\mathbf{n}_{12}$) are coplanar, while (SP2) distributes the constraint between moving objects, as if they were bound to stay on their respective side of the possibly moving hyperplane (see Appendix I).

Note that whereas (SP1) is a somewhat difficult geometric problem with non linear constraint, (SP2) is a standard minimization problem with quadratic criterion and linear constraint.

We can easily give precedence to one of the arms by using priority coefficients. We just replace the minimization term of (SP) or (SP2) by the expression $p_1 ||\mathbf{v}_1-\underline{v}_1||^2 + p_2 ||\mathbf{v}_2-\underline{v}_2||^2$, where p_1 and p_2 are positive weighting coefficients. If they are taken equal, there will be no a priori precedence given to one object. If $p_1>p_2$ the burden of avoiding collision will fall more heavily on S_2. If S_1 is a stationary obstacle, or else a moving object which cannot modify its trajectory, p_1 must be taken much larger than p_2. We will thus keep $v_1=\underline{v}_1$, while we minimize $||v_2-\underline{v}_2||$ under the constraint $(v_2-\underline{v}_1).\mathbf{n}_{12}^* \geq 0$. In this particular case, we can choose the constraint hyperplane to be in contact with S_1 at point $\mathbf{x}_1$, moving with velocity $\underline{v}_1$. Notice that for priorities to be properly defined, velocities must be normalized, this is, divided by their maximum value.

- example 1: the case of two translating spheres -

In the particular case of S_1 and S_2 being two translating spheres (in the three dimensional case) or discs (in the plane) of radius r_1 and r_2, the condition of separation of the objects can more simply be written $\mathbf{u}_{12}.\mathbf{n}_{12} \geq a_{12}$, with $\mathbf{u}_{12}$ the unit vector of the line joining the centers of the spheres O_1 and O_2, and $a_{12} = (r_1+r_2) / ||O_1O_2||$ (see Figure 2).

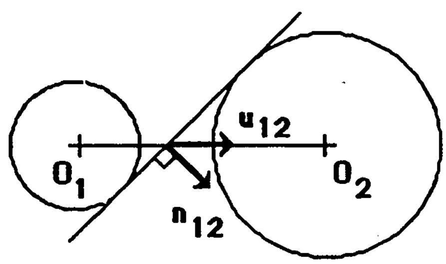


Fig. 2.

Thus subproblem (SP1) also writes:

maximize $\underline{v}_r.\mathbf{n}_{12}$ with the constraint

$\mathbf{u}_{12}.\mathbf{n}_{12} \geq a_{12}$.

This implies the optimal normal lies in the plane generated by the desired relative velocity $\underline{v}_r$ and the line joining the centers of the sphere, as illustrated in Figure 3.

With $p_{u12}(\underline{v}_r) = v_r - [v_r.u_{12}]\, u_{12}$ the projection of $\underline{v}_r$ onto the hyperplane of normal u_{12}, we derive the value of the optimal normal when the constraint is saturated: $n_{12}^* = a_{12}\, u_{12} + \sqrt{1-a_{12}^2}\; p_{u12}(\underline{v}_r)/\|p_{u12}(\underline{v}_r)\|$.

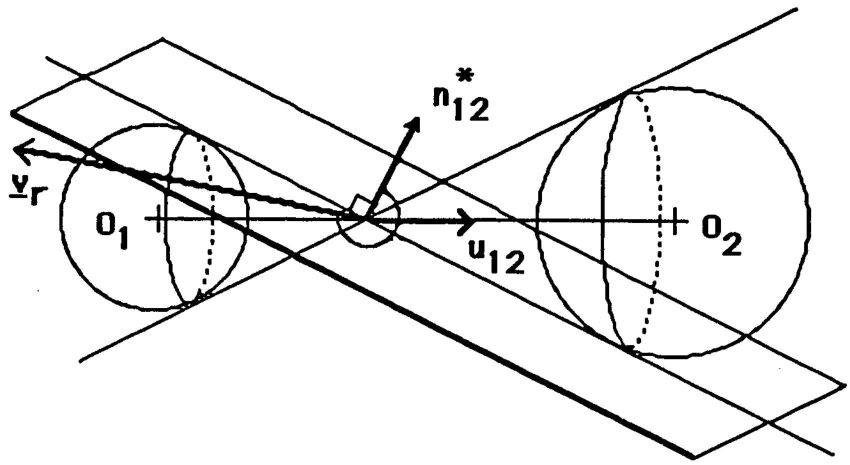


Fig. 3.

- example 2: the case of many spheres in translation -

The aim here is to coordinate the motion of n spheres in 3-D space (or discs in the plane). We write the global optimization problem of variables $\mathbf{v}_i$ and $\mathbf{n}_{ij}$, normals to the separating hyperplane for spheres S_i and S_j:

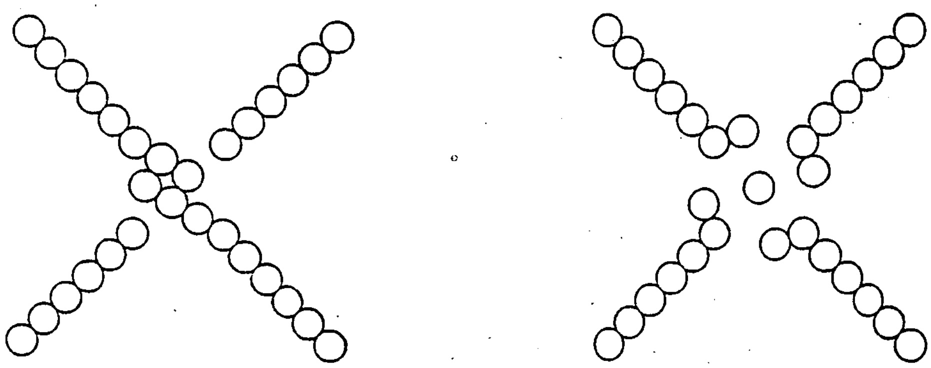
(P) minimize $\Sigma_{i=1...n}\, p_i \|\mathbf{v}_i - \underline{\mathbf{v}}_i\|^2$ with the constraints

$\forall$ (i,j), i < j, (i) $\mathbf{u}_{ij} . \mathbf{n}_{ij} \geq a_{ij}$

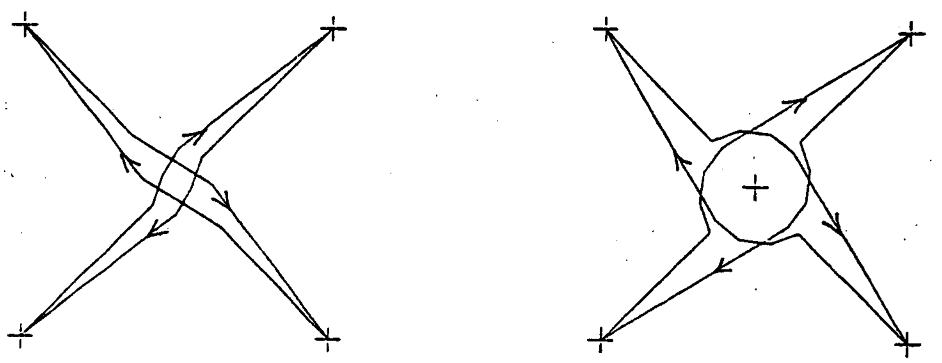
(ii) $(\mathbf{v}_j - \mathbf{v}_i).\mathbf{n}_{ij} \geq 0$.

Unfortunately we cannot uncouple problem (P) easily, the reason being that all constraints are not saturated at the same time. This neverthless gives a worthy algorithm for the motion coordination of many spheres or discs.

In Figure 4, we present some results of a simulation of the coordination of many discs in the plane (in this case, there exists only two possible tangent separating lines for each pair of discs).



First steps



Trajectories of the centers of the discs

(a) (b)

Fig.4. On the left (a), discs at opposite vertices of the square exchange their positions: two robots wait that the others have passed. On the right (b), with a stationary obstacle at the center, they adopt the turnabout strategy.

We can think of using variable priority coefficients to get out of possible dead-locks. For example, in the case a sphere is blocked inside a set of other mobile objects that already have reached their objectives, it can augment its priority coefficient so as to temporarily push them away, and get out of this trap.

2.2. The Case of Solids Translating and Rotating

We write $\mathbf{q}_i$ the configuration parameters of S_i and $\dot{\mathbf{q}}_i = \partial \mathbf{q}_i/\partial t$ their derivatives. The first idea would be to write the optimization program:

(P) minimize $p_1 \|\dot{\mathbf{q}}_1 - \dot{\underline{\mathbf{q}}}_1\|^2 + p_2 \|\dot{\mathbf{q}}_2 - \dot{\underline{\mathbf{q}}}_2\|^2$ with the constraints

(i) $[\mathbf{x}_2(\mathbf{n}_{12}) - \mathbf{x}_1(\mathbf{n}_{21})].\mathbf{n}_{12} \geq 0$

(ii) $[\mathbf{v}(\mathbf{x}_2) - \mathbf{v}(\mathbf{x}_1)].\mathbf{n}_{12} \geq 0$

Again underlined values $\dot{\underline{\mathbf{q}}}_i$ stand for *desired* values of configuration parameters derivatives. Here we must be aware that velocities depend on the points of the solids for which they are calculated, which themselves depend on the choice of the normal to the separating hyperplane. Nevertheless the following proposition states that constraints (i) and (ii) ensure that there will be no collision between the two moving objects.

Proposition 3: *We will generate no overlap between moving objects if we impose condition* $\mathbf{v}_r.\mathbf{n}_{12} \geq 0$ *at each time* t, *where:*

- $\mathbf{n}_{12} = \mathbf{n}_{12}(t)$ *is the normal to some separating hyperplane, a continuous function of time,*
- $\mathbf{v}_r = \mathbf{v}(\mathbf{x}_2) - \mathbf{v}(\mathbf{x}_1)$ *is the relative velocity calculated at the nearest points of* S_1 *and* S_2 *in direction* $\mathbf{n}_{12}$, $\mathbf{x}_2(\mathbf{n}_{12})$ and $\mathbf{x}_1(\mathbf{n}_{21})$.

Proof (see Figure 5)

Let c_2 be the point describing the trajectory of the first point of S_2 in the direction $\mathbf{n}_{12}$, and x_2 the point bound to S_2 coinciding with c_2 at time t. Let c_1 and x_1 be the equivalent points for S_1 in direction $\mathbf{n}_{21}$. We will generate no overlap if we respect condition $[v(c_2)-v(c_1)].\mathbf{n}_{12} \geq 0$ at each time t.

We have $v(c_2) = v(c_2)^{S2} + v(x_2)$, where $v(c_2)^{S2}$ is the velocity of c_2 in a frame bound to S_2. Since c_2 describes a trajectory on the surface of solid S_2, $v(c_2)^{S2}.\mathbf{n}_{12}=0$. From this and the symetrical equation for object S_1, we derive that conditions $[v(c_2)-v(c_1)].\mathbf{n}_{12} \geq 0$ and $v_r.\mathbf{n}_{12} \geq 0$ are equivalent. •

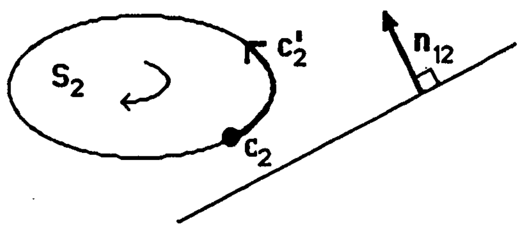


Fig. 5.

Yet we obtain a better coordination of movements if we force both objects to slide on the hyperplane with a larger velocity than what is given by problem (P), this is, if we compute the constraints as far as possible from the axes of rotation. For this, we use subprograms (SP1) and (SP2) with a criterion in (SP1) forcing the projection of the desired relative velocity to be as large as possible. We write:

(SP1) maximize $||p_{n12}(\underline{v}_r)||$ with the constraint

$$[x_2(\mathbf{n}_{12})-x_1(\mathbf{n}_{21})].\mathbf{n}_{12} \geq 0,$$

where $p_{n12}(\underline{v}_r) = \underline{v}_r - [\underline{v}_r.\mathbf{n}_{12}]\ \mathbf{n}_{12}$, with $\underline{v}_r = \underline{v}(x_2)-\underline{v}(x_1)$.

(SP2) minimize $p_1\ ||\dot{q}_1-\dot{\underline{q}}_1||^2 + p_2\ ||\dot{q}_2-\dot{\underline{q}}_2||^2$ with the constraint

$$[v(x_2)-v(x_1)].\mathbf{n}_{12}^* \geq 0,$$

where $\mathbf{n}_{12}^*$ is the optimal normal given by (SP1).

3. An Anti-collision Module for Two Manipulators

3.1. Motion Coordination of Two Articulated Chains

The manipulators, named $Robot_1$ and $Robot_2$, are two open articulated chains. $Robot_i$ is composed of $N_i + 1$ bodies, $B_i^0 \ldots B_i^{Ni}$, related by translational or revolute joints. B_i^0 is the base of the robot. Joint variables are written $\mathbf{q}_1$ and $\mathbf{q}_2$, whith $q_i=[q_i^1,\ldots,q_i^{Ni}]$. We represent a body by a union of simple convex solids, called *primitives* in the sequel. At each time t, the interaction between the two manipulators is described by a list of pairs of primitives lying at a distance less than a threshold, one on each robot, that need to be separated.

We can then use the formalism introduced above. For each pair of the list, a subproblem (SP1) is solved so as to find the best separating plane normal $\mathbf{n}_{12}^c$, $c=1\ldots N_c$, with N_c the number of interacting pairs. The corresponding constraint of separation of trajectories $[\mathbf{v}(\mathbf{x}_2)-\mathbf{v}(\mathbf{x}_1)].\mathbf{n}_{12}^c \geq 0$ translates into the following expression, using $\mathbf{J}_1$ the jacobian matrix of $Robot_1$ at point $\mathbf{x}_1$ and $\mathbf{J}_2$ the jacobian matrix of $Robot_2$ at point $\mathbf{x}_2$:

$$(J_2\, \dot{q}_2).\mathbf{n}_{12}^c - (J_1\, \dot{q}_1).\mathbf{n}_{12}^c \geq 0.$$

With $\mathbf{v}_2^c = \mathbf{J}_2^t\, \mathbf{n}_{12}^c$ and $\mathbf{v}_1^c = \mathbf{J}_1^t\, \mathbf{n}_{21}^c = -\mathbf{J}_1^t\, \mathbf{n}_{12}^c$, we derive the corresponding linearized constraint:

$$\dot{q}_1.\mathbf{v}_1^c + \dot{q}_2.\mathbf{v}_2^c \geq 0.$$

Once all optimal separating hyperplanes have been found for the N_c pairs of interacting primitives, we solve the problem of the choice of optimal joint increments with subproblem:

(SP2) minimize $p_1\, ||\dot{q}_1-\dot{\underline{q}}_1||^2 + p_2\, ||\dot{q}_2-\dot{\underline{q}}_2||^2$ with the constraints

$$\dot{q}_1 . \mathbf{v}_1^c + \dot{q}_2 . \mathbf{v}_2^c \geq 0, \text{ for } c = 1 \ldots N_c,$$

$$-\omega_1^{k1} \leq \dot{q}_1^{k1} \leq \omega_1^{k1} \text{ for } k_1 = 1 \ldots N_1,$$

$$-\omega_2^{k2} \leq \dot{q}_2^{k2} \leq \omega_2^{k2} \text{ for } k_2 = 1 \ldots N_2.$$

The last two blocks of inequations express bounds on joint velocities. Again (SP2) can be solved easily, since it is a minimization problem with quadratic criterion and linear constraints.

Yet we have done nothing so far to ensure the compatibility of the constraints chosen for the different pairs of solids that need to be separated. Indeed, in the case of two open articulated chains, it is better to first separate bodies situated at the extremities of the chains, B_1^{N1} and B_2^{N2}, and then propagate the corresponding constraint. To realize this, we use the following heuristic, that we will illustrate with the example of two vertical robots, each one with three revolute joints.

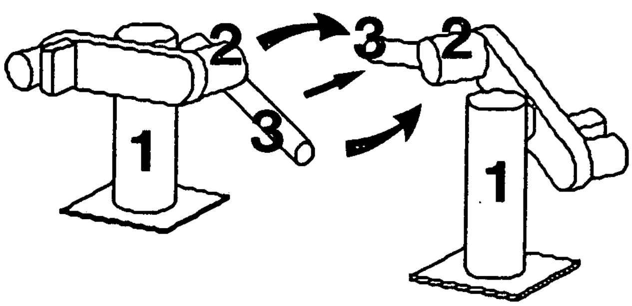


Fig. 6. Improving the compatibility of the constraints

If a primitive on the i-th body of Robot_1 and one on the j-th member of Robot_2 need to be separated we say we generate a [i,j] interaction. For example, if we generated a [2,3] and a [3,2] interaction, we first saturate possible interactions by forcing a [3,3] interaction, this is an interaction at the extremities of the articulated chains. We now compute the best separating plane for the *deepest* interaction, here [3,3], and then add an attractive term towards corresponding normal in the criteria used for the computation of normals for other interactions.

3.2. General Description of the Anti-collision Algorithm

In this application the robotic cell is composed of two manipulators sharing common workspace. At each time t, we assume to have knowledge of the joint variables of both robots and of desired joint increments, computed independentely for each robot as if there were no other one. Given joint variables and the available CAD model of the robots, we compute the list of pairs of primitives that need to be separated at time t, this is, primitives lying at a distance less than $[v_1^{max}+v_2^{max}]\Delta t$, where v_i^{max} is the maximum cartesian velocity of points on the primitives, and Δt the anticipation time. Actually, we only consider pairs of primitives from bodies whose *Maximum Swept Volume* intersect, the list being preprocessed once and for all. Here the *Maximum Swept Volume* of a body is defined as the set of points of real space that can be reached by any point of the primitive when joint angles vary freely within mechanical bounds.

In [Faverjon and Tournassoud 1986] we give algorithms for fast *intersection* and *interaction* tests between two manipulators of general architecture. They are based on the hierarchical model of solids given in [Faverjon 1986], plus a hierarchical representation of the swept volumes of the bodies of manipulators.

In [Tournassoud 1986], we in particular present ideas for reducing the number of pairs in the interaction list. An *envelope* of a moving object between time t and t+Δt is a volume of real space containing at least all the points that it can reach during that time from its present position (see Appendix II). We compute envelopes in the most restrictive way, taking into account all known limitations on future movements. We generate an interaction between two primitives describing bodies of the manipulators if and only if their envelopes intersect. Constraints can be taken into account progressively: when we first detect an intersection between two envelopes, we have at least a time Δt left during which no collision can occur between corresponding primitives.

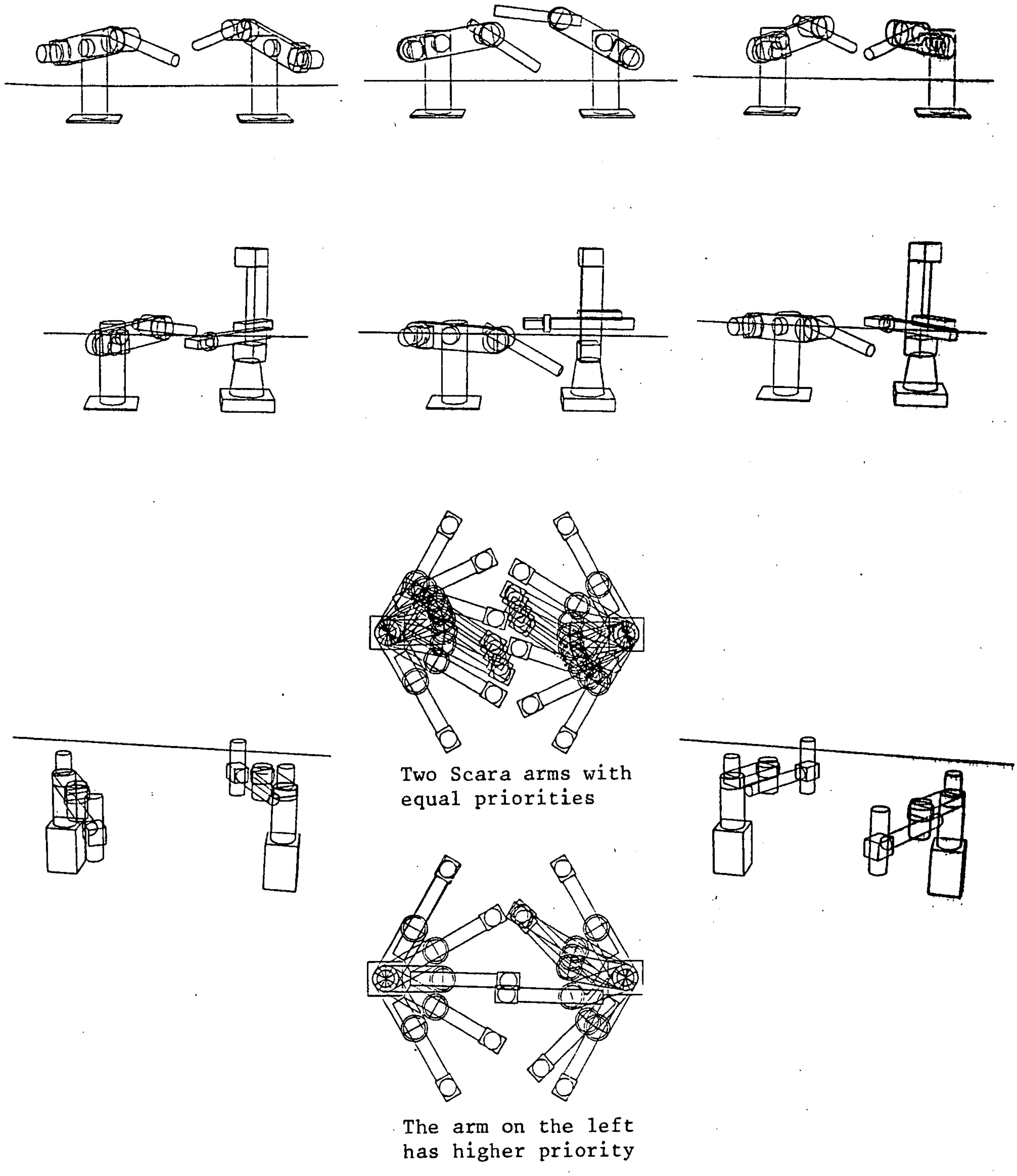
Note that a practical requirement would be that arms do not pass at less that a securiry distance, say d_{min}. This is done by modifying the constraint of (SP1), $\mathbf{x_1x_2.n_{12}}\geq 0$, into $\mathbf{x_1x_2.n_{12}}\geq d_{min}$. This is a better solution than growing the primitives, as this would arbitrarily add pairs that cannot collide to the preprocessed list of possible interactions.

4. Conclusion

As compared with others, a strong point of this approach is that it is of general application and does not rely on a specific architecture of the manipulators. Another interesting result is that we effectively take advantage of every degrees of freedom of the problem. We even can use variable priorities to augment the research space, in particular to get out of possible dead-locks. Our method compares favorably with the *Artificial Potential Field* method. Indeed, we take the problem of separation of solids as a single geometric problem, and translate anti-collision into natural constraints on the task.

A simulation based on this algorithm has been computed, using the three-dimensional representation of manipulators described in [Faverjon and Tournassoud 1986]. The execution of a trajectory (on a Perkin-Elmer 32-40) is still about 10 times slower than what it should be for real-time coordination of two arms, but the proposed general algorithm seems a worthy base for the realization of an on-line anti-collision module. For a given robot architecture, the algorithm should be simplified so as to only retain what is necessary in that case.

Figure 7 shows some views of trajectories computed with the algorithm.



Initial positions *Intermediary...* *Final...*

Fig.7.

Appendix I

The choice of the optimal normal $\mathbf{n}_{12}$ to the separating hyperplane and of optimal velocities $\mathbf{v}_i$ writes as follows, with p_i the priority coefficients, $p_i \geq 0$:

(P) minimize $p_1 \|\mathbf{v}_1-\underline{\mathbf{v}}_1\|^2 + p_2 \|\mathbf{v}_2-\underline{\mathbf{v}}_2\|^2$ with constraints

(i) $\mathbf{x}_1\mathbf{x}_2 .\mathbf{n}_{12} \geq 0$

(ii) $(\mathbf{v}_2-\mathbf{v}_1).\mathbf{n}_{12} \geq 0$

Either the absolute minimum can be reached, this is $\mathbf{v}_1=\underline{\mathbf{v}}_1$ and $\mathbf{v}_2=\underline{\mathbf{v}}_2$, or *both* constraints (i) and (ii) are saturated, which gives Proposition 1.

In the latter case, we can write that the derivatives of the Lagrangian of problem (P), $L^{(P)}$, are equal to zero, with

$L^{(P)} = [p_1 \|\mathbf{v}_1-\underline{\mathbf{v}}_1\|^2 + p_2 \|\mathbf{v}_2-\underline{\mathbf{v}}_2\|^2]/2 + \lambda \mathbf{x}_1\mathbf{x}_2 .\mathbf{n}_{12} + \mu(\mathbf{v}_2-\mathbf{v}_1).\mathbf{n}_{12} + \xi\|\mathbf{n}_{12}\|^2/2.$

From $\partial L^{(P)}/\partial \mathbf{v}_2 = p_2(\mathbf{v}_2-\underline{\mathbf{v}}_2) + \mu\mathbf{n}_{12} = 0$

and $\partial L^{(P)}/\partial \mathbf{v}_1 = p_1(\mathbf{v}_1-\underline{\mathbf{v}}_1) - \mu\mathbf{n}_{12} = 0$,

we derive $\mathbf{v}_r - \underline{\mathbf{v}}_r + \mu(p_1+p_2)/p_1p_2\, \mathbf{n}_{12} = 0$.

$\partial L^{(P)}/\partial \mathbf{n}_{12} = 0$ writes $\lambda \mathbf{x}_1\mathbf{x}_2 + \mu(\mathbf{v}_2-\mathbf{v}_1) + \xi\mathbf{n}_{12} = 0$ as $[\partial \mathbf{x}_i/\partial \mathbf{n}_{12}]^t\, \mathbf{n}_{12}$ is identically zero, since $\mathbf{x}_i$ describes a trajectory on the surface of solid S_i.

Combining the last two equations, we obtain

$\mathbf{n}_{12} = \alpha \mathbf{x}_1\mathbf{x}_2 + \beta\underline{\mathbf{v}}_r$, if $\alpha=\lambda/[2\mu^2-\xi]$ and $\beta=\mu/[2\mu^2-\xi]$.

As for problem (P), we write the lagrangian equations for subproblems (SP1) and (SP2) in the case the constraints are saturated. We recall the formulations:

(SP1) maximize $\underline{\mathbf{v}}_r.\mathbf{n}_{12}$ under the constraint of separation of solids

$\mathbf{x}_1\mathbf{x}_2 .\mathbf{n}_{12} \geq 0$ (for variable $\mathbf{n}_{12}$)

(SP2) minimize $p_1 \|\mathbf{v}_1-\underline{\mathbf{v}}_1\|^2 + p_2 \|\mathbf{v}_2-\underline{\mathbf{v}}_2\|^2$ with separation of trajectories

$(\mathbf{v}_2-\mathbf{v}_1).\mathbf{n}_{12} \geq 0$ (for variables $\mathbf{v}_1$ and $\mathbf{v}_2$)

The Lagrangians write:

$L^{(SP1)} = \underline{v}_r.\mathbf{n}_{12} + \lambda \mathbf{x}_1\mathbf{x}_2 .\mathbf{n}_{12} + \xi\|\mathbf{n}_{12}\|^2 /2$ and

$L^{(SP2)} = [p_1 \|v_1-\underline{v}_1\|^2 + p_2 \|v_2-\underline{v}_2\|^2]/2 + \mu(v_2-v_1).\mathbf{n}_{12}$.

$\partial L^{(SP1)}/\partial \mathbf{n}_{12} = 0$ gives $\mathbf{n}_{12} = \alpha \mathbf{x}_1\mathbf{x}_2 + \beta \underline{v}_r$,

while $\partial L^{(SP2)}/\partial v_i = 0$ gives $p_2 (v_2-\underline{v}_2) + \mu \mathbf{n}_{12} = 0$ and $p_1 (v_1-\underline{v}_1) - \mu \mathbf{n}_{12} = 0$.

These equations being identical from those derived from problem (P), proof of Proposition 2 is completed.

Notice that an interesting alternative is to consider priority coefficients p_i as variables of the optimization problem, thus adding one degree of freedom per moving object. In that case, we must impose the constraint $p_1+p_2=1$ on these coefficients.

Appendix II

a) Envelopes in Joint Space

We consider a manipulator with N rotational and translational degrees of freedom. We assume joint values $(q_1,\ldots,q_N)$ to be subject to the following constraints:

- maximum joint velocities $\omega_i^{max} = \max(|\dot{q}_i|)$,
- fixed signs of joint increments $\varepsilon_i = \mathrm{sign}(\dot{q}_i)$ along a trajectory from initial joint values to objectives q_i^{obj}.

These two constraints on the trajectory translate into the following description of the envelopes in joint space:

$$\min \{q_i(t) ; \max (q_i^{obj}, q_i(t) + \varepsilon_i\, \omega_i^{max}\, \Delta t)\}$$
$$\leq q_i^{env} \leq \max \{q_i(t) ; \min (q_i^{obj} , q_i(t) + \varepsilon_i\, \omega_i^{max}\, \Delta t)\}$$

b) Envelopes in Real Space

As defined above, envelopes have the interesting property to tend towards actual bodies of the manipulators when joint values tend to the objectives. Unfortunately, though the following proposition provides an algorithm for the computation of envelopes in the plane, no easy description seems to be available in the 3-D case.

Proposition: *Let* $S_i(q_1,\ldots,q_N)$ *be the surface describing the i-th body of a* two dimensional *manipulator,* Env_i *its envelope defined with above equations. If the boundary of* S_i *can be described by a finite union of arcs and segments, then the same property stands for* Env_i.

Proof: Let S be a surface whose boundary is described by a union of arcs and segments. We have the following description of the space swept by S while it is rotated or translated.

(a) *Case of a rotation.* In a frame relative to S which origin is the center of rotation, we describe S by the polar coordinates (r,α). Let $S(\theta)$ be the space occupied by S when it is rotated of an angle θ. Then the boundary of $\cup S(\theta)$ for $\theta_1 \leq \theta \leq \theta_2$ is included in the union of the boundary of $S(\theta_1)$, the boundary of $S(\theta_2)$, and of the arcs described by the local extrema of r during the sweep from θ_1 to θ_2.

(b) *Case of a translation.* In a frame relative to S which x axis is parallel to the axis of translation, we describe S by the cartesian coordinates (x,y). Then the boundary of $\cup S(x)$ for $x_1 \leq x \leq x_2$ is included in the union of the boundary of $S(x_1)$, the boundary of $S(x_2)$, and the segments described by the local extrema of y during the sweep from x_1 to x_2.

As $Env_i = \cup_{q1}\ldots\cup_{qi} S_i(q_1,\ldots,q_i)$, result follows by induction on k, k=i down to 1, considering at each step the surface $\cup_{qk}\ldots\cup_{qi} S_i(q_1,\ldots,q_i)$ •

Testing the intersection of two surfaces is equivalent to testing the intersection of their boundaries or the inclusion of the surfaces, which is done by checking if one point of one of the surface lies in the opposite surface, and respectively. Thus, the envelope intersection problem for such manipulators is reducible to intersections of arcs and segments.

References

Brooks, R. A. 1982. Solving the Find-Path Problem by Representing Free Space as Generalized Cones. A.I. Memo 674, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Brooks, R. A., and Lozano-Pérez, T. 1982. A Subdivision Algorithm in Configuration Space for Findpath with Rotation. A.I. Memo No. 684, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Erdmann, M. and Lozano-Pérez, T. 1986. On Multiple Moving Objects. Proceedings of the IEEE International Conference on Robotics and Automation, San Fransisco, pp. 1419-1424.

Faverjon, B. 1984. Obstacle Avoidance Using an Octree in the Configuration Space of a Manipulator. Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, pp. 504-512.

Faverjon, B. 1986. Object Level Programming of Industrial Robots. Proceedings of the IEEE International Conference on Robotics and Automation, San Fransisco, pp. 1406-1411.

Faverjon, B. and Tournassoud, P. 1986. A CAD System for Multi-robot Coordination. Proceedings of the NATO International Advanced Research Workshop on Languages for Sensor-based Control in Robotics, Pisa.

Fortune, S., Wilfong, G. and Yap, C. 1986. Coordinated Motion of Two Robot Arms. Proceedings of the IEEE International Conference on Robotics and Automation, San Fransisco, pp. 1216-1223.

Freund, E. and Hoyer, H. 1984. Collision avoidance for industrial robots with arbitrary motion. Journal of Robotic Systems, Vol. 1, No. 4.

Hopcroft, J.E., Schwartz, J.T. and Sharir, M. 1984. On the complexity of motion planning for multiple independent objects. The International Journal of Robotics Research, Volume 3, No. 4, pp. 76-88.

Kant, K. and Zucker, S.W. 1986. Planning Smooth Collision-free Trajectories: Path, Velocity and Splines in Free-space. Technical Report 86-2R, McGill University Computer Vision and Robotics Laboratory.

Lozano-Pérez, T. 1981. Automatic Planning of Manipulator Transfer Movements. IEEE Transactions on Systems, Man, and Cybernetics (SMC-11), pp. 681-698.

Lozano-Pérez, T. 1983. Spatial Planning: a Configuration Space Approach. IEEE Transactions on Computers (C-32), pp. 108-120.

O'Dunlaing, C. and Yap, C.K. 1983. The Voronoi Method for Motion-planning: the Case of a Disc. Technical Report No. 53, New York University Computer Science Department, Courant Institute of Mathematical Sciences.

Schwartz, J.T. and Sharir, M. 1982. On the Piano Movers' Problem II. General Techniques for Computing Topological Properties of Algebraic Manifolds. Technical Report No. 41, New York University Computer Science Department, Courant Institute of Mathematical Sciences.

Schwartz, J.T. and Sharir, M. 1983. On the Piano Movers' Problem III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Amidst Polygonal Barriers. The International Journal of Robotics Research, Volume 2, No. 3, pp. 46-75.

Spirakis, P. and Yap, C.K. 1983 (a). On the Combinatorial Complexity of Motion Coordination. Technical Report No. 76, New York University Computer Science Department, Courant Institute of Mathematical Sciences.

Spirakis, P. and Yap, C.K. 1983 (b). Strong NP-hardness of Moving Many Discs. Technical Report No. 92, New York University Computer Science Department, Courant Institute of Mathematical Sciences.

Tournassoud, P. 1986. A Strategy for Obstacle Avoidance and its Application to Multi-robot Systems. Proceedings of the IEEE International Conference on Robotics and Automation, San Fransisco, pp. 1224-1229.

Udupa, S. 1977. Collision Detection and Avoidance in Computer Manipulators. Proceedings of the 5th International Joint Conference on Artificial Intelligence, pp. 737-748.

Yap, C.K. 1984. Coordinating the Motion of Several Discs. Technical Report No. 105. New York University Computer Science Department, Courant Institute of Mathematical Sciences.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique